

Screencast: [26-kvm-intro.webm](#) or [26-kvm-intro.mp4](#)

KVM History

KVM (http://www.linux-kvm.org/page/Main_Page) stands for Kernel-based Virtual Machine and was added to the mainline Linux kernel starting with 2.6.20 (early **2007**). KVM started as a development project lead by Avi Kivity and funded by technology startup **Qumranet** in Israel.

Qumranet created a commercial product based on KVM named **SolidICE** which specialized in end-user desktop machines. SolidICE included a patented remoting protocol named **SPICE** which is similar to the RDP, Citrix ICA, Pc-over-IP, etc.

In September 2008 Qumranet was **acquired by Red Hat**.

How KVM Works

KVM requires CPU hardware support for virtualization in the CPU. It is implemented within the Linux kernel via three kernel modules... one for KVM proper (kvm) and two additional modules... one for each CPU type supported - Intel (kvm_intel) and AMD (kvm_amd). All KVM guests are "fully-virtualized".

The design behind KVM is to use the Linux kernel as the hypervisor... so you get all of the device drivers / hardware support and other functionality that is already part of the Linux kernel (scheduler, memory management, etc) "for free". By reusing code (the Linux kernel) KVM is actually very lean and less complicated when compared to other, stand-alone hypervisors who have to provide their own device driver and OS functions.

KVM is a kernel module, what about the userland?

Disk Storage

qcow2 is the native disk image format used by KVM. It is typically thin-provisioned but can be fully pre-allocated if desired.

RAW disk format is also available and fully pre-allocated

It is possible to take an existing VM disk image from a few other virt platforms and convert them but often some internal changes may be needed (think drivers) post conversion.

Emulation vs. virtio

KVM provides a good range of virtualized hardware (examples: e1000 and RTL8139 NICs) but they can be less efficient because the hypervisor has to emulate the devices (aka work harder)

virtio is a standard for device drivers where the device "knows" it is within virtualization. virtio drivers provide much better performance than emulated devices.

libvirt - Rosetta stone of virtualization?

libvirt (<http://libvirt.org/>) is a library that is released under the GNU Lesser General Public License and is primarily sponsored by Red Hat. The goal of the project is to produce a management library / abstraction layer / API for the plethora of virtualization on the market. libvirt abstracts access to Xen, QEMU, KVM, LXC (native Linux containers), OpenVZ, User Mode Linux

and even commercial products such as VMware (type1 and type2), VirtualBox, and Microsoft's Hyper-V.

virt-manager - GUI frontend for Xen or KVM VMs based on libvirt

virt-manager (<http://virt-manager.org/>) is a fairly comprehensive application that lets you create, configure, start, stop, destroy, and re-configure KVM virtual machines. It also offers performance monitoring and access to the graphical console of your VMs (via SPICE or VNC). Available for Microsoft Windows too.

It can have one or more "connections" for managing local or remote VMs of type **system** or **session** (user).

See also: **GNOME Boxes** (for session VMs)

Demo virt-manager here.

virsh -command line tool to manage Xen or KVM VMs based on libvirt

```
virsh list --all
virsh start vmname
virsh shutdown vmname
```

virsh can also be used in an interactive mode where you can actually alter the configurations of your VMs.

Demo virsh here.

virt-install - command line tool for creating virtual machines

```
virt-install --name centos74 --ram 2048 \
--disk path=/vm/demo1.img,size=10 --network=bridge:vmbr0 \
--vnc --os-type=linux --os-variant=rhel7 \
--cdrom /isos/centos7-x86_64.iso --accelerate
```

Although virt-install is a command line application, when doing an interactive OS install, you almost always need access to a graphical console. Luckily virt-install will automatically launch remote-viewer.

virt-viewer (also remote-viewer)

virt-viewer is a minimal tool for displaying the graphical console of a virtual machine. The console is accessed using the VNC protocol.

The guest can be referred to based on its name, ID, or UUID.

```
spice://host.example.com:port
(starts at 5900)
```

Other tools

Some additional tools from the libguestfs-tools package and other packages include:

cockpit-machines - add-on for web-based Cockpit control-panel, to eventually replace virt-manager

qemu-img - Work with disk images

guestmount - Mount a guest on the host

virt-cat - Display a file in a virtual machine

virt-clone - Clone an existing VM to make a new one
virt-df - Display free space on virtual filesystems
virt-edit - Edit a file in a virtual machine
virt-inspector - Display information about a VM
virt-filesystems - List filesystems in a VM or disk image
virt-ls - List filesystems in a virtual machine or disk image
virt-rescue - Run a rescue shell on a virtual machine
virt-win-reg - Display Windows Registry entries from a VM
virt-resize - Easily resize a VM disk file
virt-p2v - Convert a physical machine into a VM
virt-v2v - Convert a physical machine into a VM